

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)
[PREV](#) [NEXT](#)[FRAMES](#) [NO FRAMES](#)*Java™ 2 Platform
Standard Edition*

Java™ 2 Platform, Standard Edition, v1.2.2 API Specification

This document is the API specification of the Java 2 Platform, Standard Edition, version 1.2.2. For a searchable version of these API specifications, see [Search Engine for Documentation](#).

See:

[Description](#)

Packages	
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and an interface for the input method framework.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
java.awt.print	Provides classes and interfaces for a general printing API.
java.beans	Contains classes related to Java Beans development.

<u>java.beans.beancontext</u>	Provides classes and interfaces relating to bean context.
<u>java.io</u>	Provides for system input and output through data streams, serialization and the file system.
<u>java.lang</u>	Provides classes that are fundamental to the design of the Java programming language.
<u>java.lang.ref</u>	Provides reference-object classes, which support a limited degree of interaction with the garbage collector.
<u>java.lang.reflect</u>	Provides classes and interfaces for obtaining reflective information about classes and objects.
<u>java.math</u>	Provides classes for performing arbitrary-precision integer arithmetic (BigInteger) and arbitrary-precision decimal arithmetic (BigDecimal).
<u>java.net</u>	Provides the classes for implementing networking applications.
<u>java.rmi</u>	Provides the RMI package.
<u>java.rmi.activation</u>	Provides support for RMI Object Activation.
<u>java.rmi.dgc</u>	Provides classes and interface for RMI distributed garbage-collection (DGC).
<u>java.rmi.registry</u>	Provides a class and two interfaces for the RMI registry.
<u>java.rmi.server</u>	Provides classes and interfaces for supporting the server side of RMI.
<u>java.security</u>	Provides the classes and interfaces for the security framework.
<u>java.security.acl</u>	The classes and interfaces in this package have been superseded by classes in the java.security package.
<u>java.security.cert</u>	Provides classes and interfaces for parsing and managing certificates.
<u>java.security.interfaces</u>	Provides interfaces for generating RSA (Rivest, Shamir and Adleman AsymmetricCipher algorithm) keys as defined in the RSA Laboratory Technical Note PKCS#1, and DSA (Digital Signature Algorithm) keys as defined in NIST's FIPS-186.
<u>java.security.spec</u>	Provides classes and interfaces for key specifications and algorithm parameter specifications.
<u>java.sql</u>	Provides the JDBC package.

BEST AVAILABLE COPY

<u>java.text</u>	Provides classes and interfaces for handling text, dates, numbers, and messages in a manner independent of natural languages.
<u>java.util</u>	Contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string tokenizer, a random-number generator, and a bit array).
<u>java.util.jar</u>	Provides classes for reading and writing the JAR (Java ARchive) file format, which is based on the standard ZIP file format with an optional manifest file.
<u>java.util.zip</u>	Provides classes for reading and writing the standard ZIP and GZIP file formats.
<u>javax.accessibility</u>	Defines a contract between user-interface components and an assistive technology that provides access to those components.
<u>javax.swing</u>	Provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms.
<u>javax.swing.border</u>	Provides classes and interface for drawing specialized borders around a Swing component.
<u>javax.swing.colorchooser</u>	Contains classes and interfaces used by the JColorChooser component.
<u>javax.swing.event</u>	Provides for events fired by Swing components.
<u>javax.swing.filechooser</u>	Contains classes and interfaces used by the JFileChooser component.
<u>javax.swing.plaf</u>	Provides one interface and many abstract classes that Swing uses to provide its pluggable look-and-feel capabilities.
<u>javax.swing.plaf.basic</u>	Provides user interface objects built according to the Basic look-and-feel.
<u>javax.swing.plaf.metal</u>	Provides user interface objects built according to the "metal" look-and-feel.
<u>javax.swing.plaf.multi</u>	The multiplexing look and feel allows users to combine auxiliary look and feels with the default look and feel.
<u>javax.swing.table</u>	Provides classes and interfaces for dealing with java.awt.swing.JTable.
<u>javax.swing.text</u>	Provides classes and interfaces that deal with editable and noneditable text components.
<u>javax.swing.text.html</u>	Provides the class HTMLToolkit and supporting classes for creating HTML text editors.

BEST AVAILABLE COPY

<u>javax.swing.text.html.parser</u>	
<u>javax.swing.text.rtf</u>	Provides a class (RTFEditorKit) for creating Rich-Text-Format text editors.
<u>javax.swing.tree</u>	Provides classes and interfaces for dealing with java.awt.swing.JTree.
<u>javax.swing.undo</u>	Provides support for undo/redo capabilities in an application such as a text editor.
<u>org.omg.CORBA</u>	Provides the mapping of the OMG CORBA APIs to the Java™ programming language, including the class ORB, which is implemented so that a programmer can use it as a fully-functional Object Request Broker (ORB).
<u>org.omg.CORBA.DynAnyPackage</u>	Provides the exceptions used with the DynAny interface (InvalidValue, Invalid, InvalidSeq, and TypeMismatch).
<u>org.omg.CORBA.ORBPackage</u>	Provides the exception InvalidName, which is thrown by the method ORB.resolve_initial_references and the exception InconsistentTypeCode, which is thrown by the Dynamic Any creation methods in the ORB class.
<u>org.omg.CORBA.portable</u>	Provides a portability layer, that is, a set of ORB APIs that makes it possible for code generated by one vendor to run on another vendor's ORB.
<u>org.omg.CORBA.TypeCodePackage</u>	Provides the user-defined exceptions BadKind and Bounds, which are thrown by methods in in the class TypeCode.
<u>org.omg.CosNaming</u>	Provides the naming service for Java IDL.
<u>org.omg.CosNaming.NamingContextPackage</u>	Provides the exceptions used in the package org.omg.CosNaming (AlreadyBound, CannotProceed, InvalidName, NotEmpty, and NotFound) and also the Helper and Holder classes for those exceptions.

This document is the API specification of the Java 2 Platform, Standard Edition, version 1.2.2.

[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#)

*Java™ 2 Platform
Standard Edition*

[Submit a bug or feature](#)

Java, Java 2D, and JDBC are a trademarks or registered trademarks of Sun Microsystems, Inc. in the US and other countries.

Copyright 1993-1999 Sun Microsystems, Inc. 901 San Antonio Road,
Palo Alto, California, 94303, U.S.A. All Rights Reserved.

BEST AVAILABLE COPY